Research Journal of Applied Sciences, Engineering and Technology 8(4): 541-547, 2014
DOI:10.19026/rjaset.8.1003
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: April 01, 2014 Accepted: May 19, 2014 Published: July 25, 2014

Research Article

A Novel Optimized Adaptive Learning Approach of RBF on Biomedical Data Sets

[1]Anusuya S. Venkatesan and [2]Latha Parthiban
[1]Department of Information Technology, Saveetha School of Engineering, Saveetha University, Thiruvallur, Tamil Nadu-602105, India
[2]Pondicherry University, Puducherry, India

Abstract: In this study, we propose a novel learning approach of Radial Basis Function Neural Network (RBFNN) based on Fuzzy C-Means (FCM) and Quantum Particle Swarm Optimization (QPSO) to group similar data. The performance of RBFNN relies on the parameters such as number of hidden nodes, centres and width of Gaussian function and weight matrix between hidden layer and output layer. Generally, RBF is trained with a fixed number of nodes but in this study we allow the network to have variable number of hidden nodes based on the size of input samples. The clustering algorithm, Fuzzy C Means (FCM) is optimized with QPSO to provide global optimal centres for RBFNN. The weights are calculated by using Least Square Method and the root mean square error is optimized to improve the accuracy, accordingly the hidden unit numbers are adjusted. The cluster centres are obtained using optimized FCM and are checked against random selection of centres to verify the suitability. The datasets such as liver disorder and breast cancer from UCI machine learning repository are used for the experiments. The accuracy is analyzed for the Cluster Numbers (CN) 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 and 20, respectively.

Keywords: FCM, fitness, QPSO, RBFNN, RMSE

INTRODUCTION

RBF was proposed in the year 1988 by Broomhead and Lowe. It has been used in many applications from the inception such as speech recognition, function approximation, time series prediction etc. It has been extensively used for solving pattern recognition problems (Antonios and George, 2012; Xiao-Yuan *et al*., 2007) and also used as universal approximators in a significant number of applications (Er *et al*., 2005; Min and Jianhui, 2004). The RBFNN is an artificial network uses radial basis function as activation functions and it is a type of feed forward Neural Network (NN), has recently attracted extensive research interest because of its simple architecture, high approximation and regularization capability and good local specialization and global generalization ability. The basic architecture for a RBF is a three layer network; input, hidden and output layers. The input layer is simply a fan-out layer; pass the input vector to hidden layer. The hidden layer performs a non-linear mapping from the input space into a higher dimensional space in which the patterns become linearly separable. The most commonly used Gaussian function is the activation function of hidden nodes given in Eq. (1) (Fig. 1).

The output layer performs a simple weighted sum with a linear output:

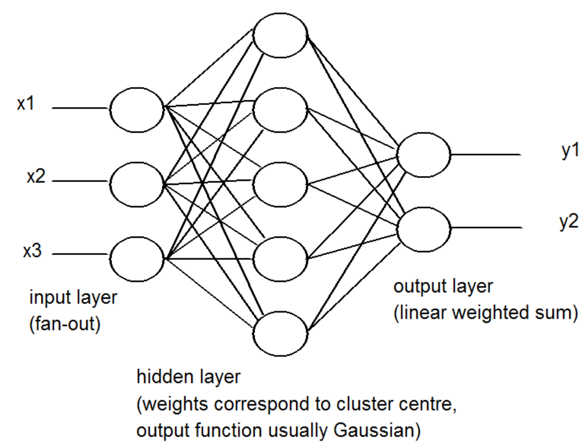

Fig. 1: Architecture of RBF

$$(hidden\_unit)\phi_j = exp\left(\frac{\|X-C_j\|^2}{2\sigma_j^2}\right), j = 1, 2, \ldots h \quad (1)$$

where, ϕ_j is the distance between input space X and centre C_j of j^{th} hidden node. The variable sigma, σ_j, defines the width or radius of RBF node. The final layer of Radial Basis network calculates weighted sum with a linear output of hidden layer. It is calculated as:

$$Y_k(X) = \sum_{j=1}^{h} \phi_j(X)^* W_{jk}, k = 1, 2, \ldots, n \quad (2)$$

Corresponding Author: Anusuya S. Venkatesan, Department of Information Technology, Saveetha School of Engineering, Saveetha University, Thiruvallur, Tamil Nadu-602105, India

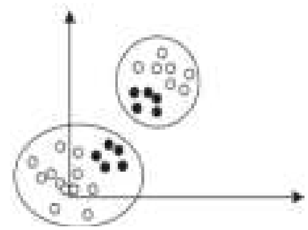

Fig. 2: Input clustering

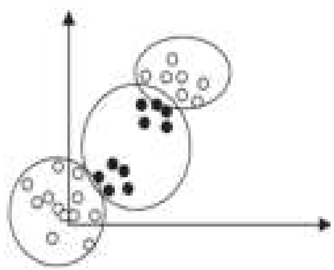

Fig. 3: Input and output clustering

where,

$Y_k(X)$ = The output of the k^{th} neuron

W_{jk} = The weight from the j^{th} neuron of the hidden layer to the k^{th} neuron of the output layer

RBF initialization and related work: RBFNN construction and learning stages involves two stages: RBFNN structure initialization and parameter optimization. In the stage of structure initialization, the number of hidden neurons is to be identified. The better structure initialization leads to good accuracy of the networks. Generally, clustering algorithms such as K-means, FCM (Grisales *et al*., 2004) determines initial structure of RBF. These algorithms work in two modes; supervised learning and unsupervised learning. In case of unsupervised learning process, the algorithm considers only input space for clustering (Harun *et al*., 2012) but in supervised mode, considers both input and output space (Xin-Zheng *et al*., 2012). The traditional unsupervised clustering methods group data into clusters based on the predefined number of clusters. The wrong number of clusters may reduce system accuracy. The following Fig. 2 shows the clustering result of standard algorithm without considering the output and Fig. 3 shows the result of clustering while considering both input and output. Figure 3 ensures for proper clustering.

Supervised learning methods consider both input and output for better system initialization. The output function is written as O = (XWY) where X is the input vector, Y is the output vector and W is the weight.

In general, hybridization of algorithms is used extensively in many applications for better performance by exploiting good properties and applying those properties in standard algorithm to mitigate its weakness.

Optimization algorithms were employed for system initialization and parameter optimization of RBF. Genetic algorithm, PSO, Differential evolution is popular optimization methods hybridized with RBF for better performance. PSO is used to train RBF and to find optimal parameters for fuzzy clustering (George and Tsekouras, 2013). A hybrid approach combining PSO and RBF is proposed to solve classification problems (Leung *et al*., 2012). A new Optimum Steepest Decent (OSD); a combination of PSO and gradient decent proposed to initialize RBF more accurately and interesting outcomes are found (Vahid and Gholam Ali, 2013).

Clustering investigates data distribution of domain and forms similar data into different groups. Classification quantifies the relationships among the data and separates them into various classes. Generally Classification techniques are non-parametric methods. The integrated approach of clustering and classification are used for better performance. The output of hierarchical clustering is fed into classification methods such as decision trees and regression analysis to understand and characterize petroleum reservoirs (Denis *et al*., 2011). In some applications, learning based clustering approaches are used to discover the knowledge of domain. Hybrid nature of clustering and classification has been adopted to identify road accidents in Iran (Mahdi *et al*., 2013). Clustering algorithms are used often in Neural Networks (NN) to find the function centres. A novel learning strategy comprising fuzzy clustering and RBF is introduced to find compact and accurate clusters (Antonios and George, 2012). A new clustering approach with a delayed connection (Adibi *et al*., 2005) in NN architecture is proposed to obtain high precision clusters. RBF has been widely used in real time classification problem. For better training in less number of iterations, a PSO based approach is demonstrated and showed interesting outcome (Vahid and Gholam Ali, 2013). To train RBFNN, a novel Fuzzy C-Means clustering is described for effective outcome (Antonino *et al*., 2006). Learning ability of algorithm is improved by employing maximum entropy based RBF system to control chaotic system (Liu *et al*., 2006). In this study, we propose an approach to design an adaptive RBF to cluster and classify the data of biomedical data sets. The design ensures for less error and good accuracy in less computation time.

MATERIALS AND METHODS

Optimization in RBF design: Swarm intelligence algorithms also referred as nature-inspired algorithms are motivated by social behavior of animals and have proved to be very efficient in solving real world optimization problems. These algorithms include Particle Swarm Optimization (PSO), Ant Colony Optimization, Stochastic Diffusion Search and Bacteria Foraging and so on. Quantum Particle Swarm Optimization (QPSO) is a variant of PSO (Kennedy and

Eberhart, 1995) and is proposed for better search ability and good convergence speed. Inception of Quantum theory into PSO brought a new algorithm called QPSO was proposed by Sun *et al*. (2004). QPSO guaranteed optimal solution, unlike PSO no velocity vectors are needed and takes less number of parameters to adjust. QPSO has been used in wide range of optimization problems (Leandro dos Santos, 2010; Debao *et al*., 2012; Bo and Jiulun, 2008; Jun *et al*., 2012). In this study we apply QPSO to optimize the root mean square error and function centres of the network and to tune the network for adaptive nature. In PSO, particles in the swarm fly around in D dimensional space and interact with each other for optimal solution. Each particle in the population iteratively discovers its own position according to velocity and its previous positions. The individual best positions of particles are stored in order to find global best positions in each iteration. The positions of i^{th} particle in D dimensions is represented with position vector Xi = (x_{i1}, x_{i2}, x_{i3}, …, x_{in}) given in Eq. (4) and velocity vector Vi = (v_{i1}, v_{i2}, v_{i3}.. v_{in}) given in Eq. (3) The optimal solution of the problem is determined according to fitness function:

$$V_i(t+1) = \omega V_i(t) + c_1 rand()(Pbest(t) - X_1(t) + C_2 rand()(Gbest)(t) - X_i(t)) \quad (3)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (4)$$

Each particle maintains two positions:

- Personal best position (P best) is the best position experienced so far
- Global best position (G best) is the best position over entire swarm

The Eq. (3) and (4) are updated iteratively based on P best and G best values where ω is the inertia weight controls the speed of algorithm. The local and social search of PSO influenced by two positive parameters c_1 and c_2, generally set between the range (0, 2). As mentioned above QPSO does not require velocity vectors and the introduction of mean best position (mbest) increase search scope of particles, thus making the algorithm more efficient. The particles positions are calculated using Eq. (5):

$$X_i(t+1) = p \pm \beta |m\,best - X_i(t)|^* ln(1/u) \quad (5)$$

where,

$$p = rand(0,1)^* Pbest + (1 - rand(0,1))^* Gbest \quad (6)$$

And Eq. (7) shows the calculation of M best position which is the average of P best positions:

$$mbest = (1/M \sum_{i=1}^{M} Pbest_{i,1} / M \sum_{i=1}^{M} Pbest_{1,2,\dots,1} / M \sum_{i=1}^{M} Pbest_{i,D}) \quad (7)$$

An elite learning optimization approach (Leung *et al*., 2012), RBFNN using ALPSO proposed for classification problem. Velocity vectors of PSO are updated with linearly decreasing inertia weight and the proposed structure optimizes weight matrix and other controlling parameters. A PSO based optimized learning method used to Vahid and Gholam Ali (2013) optimizes RBF unit centres. Different architectures using PSO such as RBF1, RBF2, RBF3 and RBF4 (George and Tsekouras, 2013) are discussed and result shows that the number of hidden node increases as the values of fitness function decreases. PSO-RBFNN structure proposed for Electro Cardiogram (ECG) beat classification (Mehmet and Berat, 2010). A method of FCM optimized using Ant Colony Optimization (ACO) and Genetic Algorithm (GA) proposed to provide optimal RBF centres to increase accuracy rate (Zhide *et al*., 2009; Zhen *et al*., 2008). Similarly many optimization algorithms are tested with RBFNN. But very fewer papers talk about optimization of RBF using quantum techniques. In this study, a novel structure FCM-RBF-QPSO is discussed to optimize RBFNN and simulation results exhibits its superiority.

Proposed work: Generally, RBFNN training depends on three parameters such as RBF unit centres, width of RBF unit and weight matrix. RBF neuron centre usually determined either by applying clustering algorithms in the input space or choosing centres randomly. Some of the clustering algorithms used are K-means, Fuzzy C Means, Kohonen-SOM and Orthogonal Least Squares (OLS). In this study we apply FCM optimized using QPSO to find the centres of RBF.

Calculate centres and identify the number of hidden nodes: The performance of network largely depends on cluster centres. Thus, in this study much attention is given to choosing neuron centre; we apply optimization technique QPSO to minimize the objective function given in Eq. (11) to produce optimal centres. The cluster centre determines suitable number of hidden nodes. The covers theorem says the number of RBFNN units is to be bigger than the dimension of input space. Otherwise, over smoothing or over adaptation may reduce the performance. Here, we try to propose an optimal number of hidden nodes using QPSO:

$$j_{fcm} = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik}) \; \|X_k - v_i\|^2 \quad (8)$$

where,

$$v_i = \frac{\sum_{k=1}^{n} x_k (\mu_{ik})^m}{(\mu_{iK})^m} \quad (9)$$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} \frac{\|x_k - v_i\|}{\|x_k - v_j\|}} 2/(m-1) \quad (10)$$

$$J_{obj} = \min (J_{fcm}) \quad (11)$$

where,
x_k = The input vector
m = The fuzziness parameter
v_i = The centre of cluster i
μ_{ik} = The membership grade

The width of j^{th} RBF unit given in Eq. (12) and is calculated using the method proposed by Moody and Darken (1989):

$$\sigma_i = \alpha d_{max} / \sqrt{CN} \quad (12)$$

where,
d_{max} = The maximum distance between clusters
α = The positive factor and is set as 2

The output of the network is given in Eq. (2) and the standard error is calculated as:

$$J_{STE} = \sum_{k=1}^{n} \|Y - Y\|^2 \quad (13)$$

$$Y = HW \quad (14)$$

where, H = Y_k (X) given in Eq. (2).
The optimal weights are obtained using Eq. (14):

$$W_{opt} = [HTH]^{-1} HTY \quad (15)$$

Algorithm of proposed method:

1. Initialize the swarm with random positions. Set number of clusters (CN = 2)
2. Optimize the objective function of FCM using QPSO
   2.1 Initialize the swarm and its positions
   2.2 Compute Fitness and update the positions of particles as per Eq. (5), (6) and (7)
   2.3 Set gbest positions of particles as cluster centres
3. Find hidden nodes and width of RBF network using Eq. (12)
4. Sort RBF nodes according to distance d
5. Find weight matrix and optimize Root Mean Square Error (RMSE) using QPSO
6. If output is satisfactory then halt
   Else CN = CN+1
   Goto step 2

RESULTS AND DISCUSSION

The experiments were performed on the datasets from UCI machine learning repository. The characteristics of datasets are given in Table 1.

In the proposed RBFNN, the centres are determined by Optimized FCM (OFCM) and also through random selection. The fitness function or objective function given in Eq. (8) and (12) are to be minimized to improve the structure of RBF NN. The position of particles in a swarm such as Pbest and Gbest gives fitness values. OFCM produces fuzzy clusters and crisp clusters are found by random selection. Too many or very few clusters does not guarantee for good outcome. Hence, an adaptive approach is proposed in this study to choose right number of clusters based on input space. The execution results on the dataset liver disorder is shown in Table 2 while Table 3 depicts execution on Breast cancer. In this study QPSO is

Table 1: The descriptions of dataset

Data set	Number of rows	Number of columns	Description
Liver disorder	345	7	The variables in the datasets depict the blood test of male individual.
Breast cancer	699	10	The data concern with the chronological grouping of the clinical data from patients.

Table 2: Analysis of RMSE, D_{max} on the dataset liver disorder

CN	RMSE training/testing	D_{max} (OFCM)	D_{max} (random)	Width (OFCM/random)
3	0.2428/0.2434 (OFCM) 0.2447/0.2440 (random)	0.6132	0.4568	0.5967/0.5871
6	0.2416/0.2426 (OFCM) 0.2430/0.2418 (random)	0.5323	0.6182	0.5323/0.5048
8	0.2395/0.2366 (OFCM) 0.2408/0.2412 (random)	0.7444	0.6265	0.5264/0.4430
10	0.2322/0.2308 (OFCM) 0.2343/0.2401 (random)	1.0453	0.8502	0.6611/0.5377
15	0.2219/0.2289 (OFCM) 0.2316/0.2296 (random)	1.0144	0.9889	0.5238/0.5107
20	0.2178/0.2193 (FCM) 0.2245/0.2198 (random)	1.2326	1.1562	0.4978/0.5678

Table 3: Analysis of RMSE, D_{max} on the dataset breast cancer

CN	RMSE training/testing	D_{max} (FCM)	D_{max} (random)	Width (OFCM/random)
3	0.0343/0.1219 (OFCM) 0.0385/0.7094 (random)	1.3196	1.7424	1.5238/2.0119
6	0.0259/0.0824 (OFCM) 0.0403/0.0942 (random)	1.6729	2.2280	1.3659/1.8191
8	0.0232/0.0530 (OFCM) 0.0359/0.0824 (random)	1.6729	1.3848	1.1307/1.8191
10	0.0215/0.0433 (OFCM) 0.0386/0.0849 (random)	2.3974	2.3824	1.5067/1.5162
15	0.0193/0.0146 (OFCM) 0.0378/0.0780 (random)	1.9428	1.9705	1.0033/1.0175
20	0.0270/0.0236 (OFCM) 0.1845/0.0657 (random)	2.5200	2.4180	1.1270/1.0814

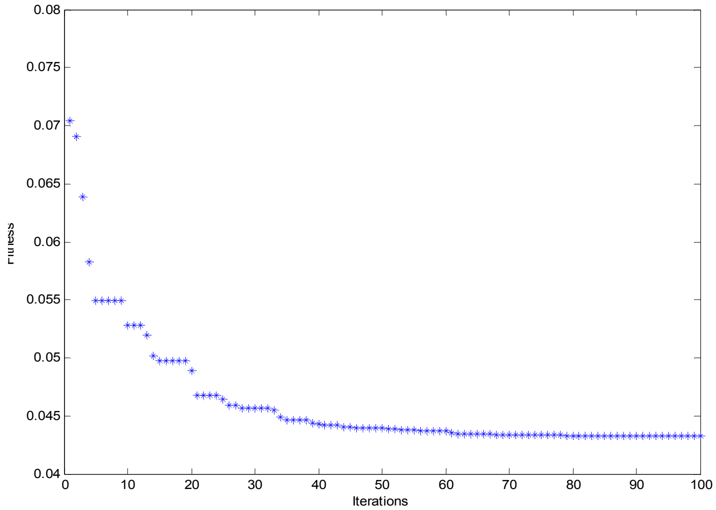

Fig. 4: Fitness values for number of hidden nodes = 8

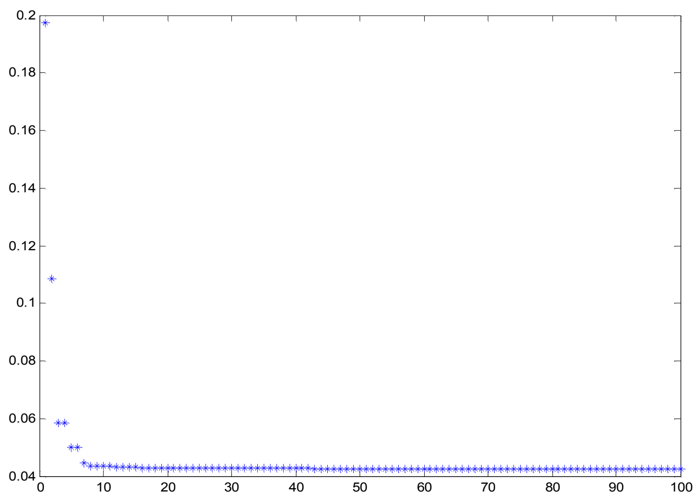

Fig. 5: Fitness values for number of hidden nodes = 10

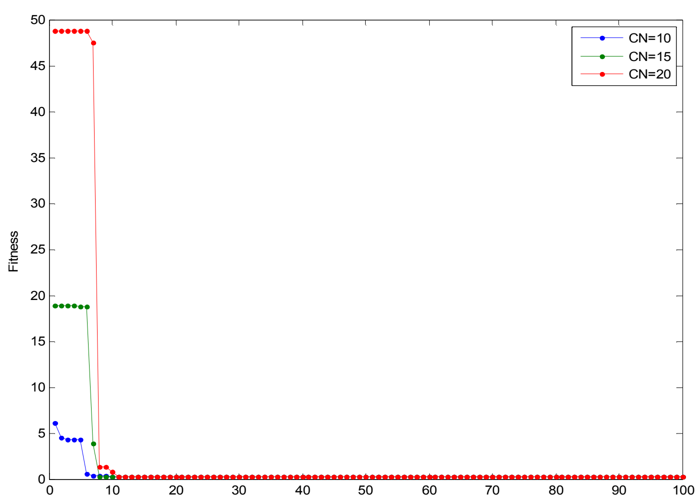

Fig. 6: Fitness values for different number of clusters on the dataset liver disorder

used as a multi objective optimization method since it optimizes centres and cost function simultaneously. The Root Mean Square Error (RMSE) is calculated for both cases on applying OFCM and choosing random centres from the input. The analysis is also carried out on $D_{max;}$ the distance between the centres. From Table 2 and 3, the RMSE on training and testing decreases as the number of cluster increases.

Figure 4 to 6 shows the fitness of cost function. The fitness decreases when the number of hidden nodes increases.

Figure 4 to 6 shows the fitness evaluation of cost function on liver disorder and breast cancer datasets. The proposed method produces the fitness for cluster numbers CN = 10, 15 and 20 as 0.2468, 0.24413 and 0.2211 respectively at 100^{th} iteration.

CONCLUSION

In this study we propose a supervised self adaptive RBF neural network to cluster the data sets efficiently. The network suffers with slow training, wastage of memory and less accuracy if correct number of hidden neurons is not chosen. Our approach identifies the suitable number of clusters for the data sets with reduced root mean square error. The prominent features of network such as good convergence speed and high precision are obtained by the proposed method. The inclusion of optimization technique QPSO improves the performance of the network. This supervised self adaptive network produce lower error for training and testing data when OFCM is executed on input space and the suitable number of hidden nodes automatically set by network from previous learning.

REFERENCES

Adibi, P., M.R. Meybodi and R. Safabakhsh, 2005. Unsupervised learning of synaptic delays based on learning automata in an RBF-like network of spiking neurons for data clustering. Neurocomputing, 64: 335-357.

Antonino, S., T. Roberto and P. Witold, 2006. Improving RBF networks performance in regression tasks by means of a supervised fuzzy clustering. Neurocomputing, 69(13-15): 1570-1581.

Antonios, D.N. and E.T. George, 2012. A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach. Fuzzy Set. Syst., 193: 62-84.

Bo, L. and F. Jiulun, 2008. Parameter selection of generalized fuzzy entropy-based thresholding method with quantum-behavior particle swarm optimization. Proceeding of the International Conference on Audio, Language and Image Processing, pp: 546-551.

Debao, C., W. Jiangtao, Z. Feng, H. Weibo and Z. Chunxia, 2012. An improved group search optimizer with operation of quantum-behaved swarm and its application. Appl. Soft Comput., 12(2): 712-725.

Denis, F., L. Evelina, G. Giacomo, F. Michele and D. Raffaele, 2011. Integrating clustering and classification techniques: A case study for reservoir facies prediction. Stud. Comput. Intell., 369: 21-34.

Er, M.J., W. Chen and S. Wu, 2005. High-speed face recognition based on discrete cosine transform and RBF neural networks. IEEE T. Neural Networ., 16(3): 679-6915.

George, E. and J. Tsekouras, 2013. On training RBF neural networks using input-output fuzzy clustering and particle swarm optimization. Fuzzy Set. Syst., 221: 65-89.

Grisales, V.H., J.J. Soriano, S. Barato and D.M. Gonzalez, 2004. Robust agglomerative clustering algorithm for fuzzy modeling purposes. Proceeding of the American Control Conference. Boston, 2: 1782-1787.

Harun, P., E. Burak, P. Andy and Y. Çetin, 2012. Clustering of high throughput gene expression data. Comput. Oper. Res., 39(12): 3046-3061.

Jun, S., C. Wei, F.Wei, W. Xiaojun and X. Wenbo, 2012. Gene expression data analysis with the clustering method based on an improved quantum-behaved particle swarm optimization. Eng. Appl. Artif. Intel., 25(2): 376-391.

Kennedy, J. and R.C. Eberhart, 1995. Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Network. Perth, WA, 4: 1942-1948.

Leandro dos Santos, C., 2010. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst. Appl., 37(2): 1676-1683.

Leung, S.Y.S., T. Yang and W.K. Wong, 2012. A hybrid particle swarm optimization and its application in neural networks. Expert Syst. Appl., 39(1): 395-405.

Liu, F., C. Sun, W. Si-Ma, R. Liao and F. Guo, 2006. Chaos control of ferroresonance system based on RBF-maximum entropy clustering algorithm. Phys. Lett. A, 357(3): 218-223.

Mahdi, A., N. Ali Nedaie and A. Alimohammad, 2013. Presentation of clustering-classification heuristic method for improvement accuracy in classification of severity of road accidents in Iran. Safety Sci., 60: 142-150.

Mehmet, K. and D. Berat, 2010. ECG beat classification using particle swarm optimization and radial basis, function neural network. Expert Syst. Appl., 37(12): 7563-7569.

Min, H. and X. Jianhui, 2004. Efficient clustering of radial basis perceptron neural network for pattern recognition. Pattern Recogn., 37(10): 2059-2067.

Moody, J. and C.J. Darken, 1989. Fast learning in network of locally-tuned processing units. Neural Computation, 1: 281-294.

Sun, J., W.B. Xu and B. Feng, 2004. A global search strategy of quantum-behaved particle swarm optimization. Proceeding of the IEEE Conference on Cybernetics and Intelligent Systems, pp: 111-116.

Vahid, F. and M. Gholam Ali, 2013. An improvement in RBF learning algorithm based on PSO for real time applications. Neurocomputing, 111: 169-176.

Xiao-Yuan, J., Y. Yong-Fang, Z. David, Y. Jing-Yu and L. Miao, 2007. Face and palmprint pixel level fusion and Kernel DCV-RBF classifier for small sample biometric recognition. Pattern Recogn., 40(11): 3209-3224.

Xin-Zheng, X., D. Shi-Fei, S. Zhong-Zhi and Z. Hong, 2012. Optimizing radial basis function neural network based on rough sets and affinity propagation clustering algorithm. Comput. Electr., 13(2): 131-138.

Zhen, Z., Z. Wang, Y. Hu and M. Geng, 2008. Learning method of RBF network based on FCM and ACO. Proceeding of the Chinese Control Decision Conference, pp: 102-105.

Zhide, L., C. Jiabin and S. Chunlei, 2009. A new RBF neural network with GA-based fuzzy C-means clustering algorithm for SINS fault diagnosis. Proceeding of the Chinese Control and Decision Conference, pp: 208-221.